

Subverting the Bazaar: a hitchhiker's guide to distributed version control for SVN users

Florian Rathgeber (f.rathgeber10@imperial.ac.uk)

Applied Modeling and Computation Group (AMCG)
Department of Earth Science & Engineering
Imperial College London

AMCG Seminar
January 21st 2010

Outline

Version Control System Concepts

Collaborative Software Development Workflows

Bazaar Basics

Bazaar in an SVN environment

Outline

Version Control System Concepts

- Centralised Version Control

- Distributed Version Control

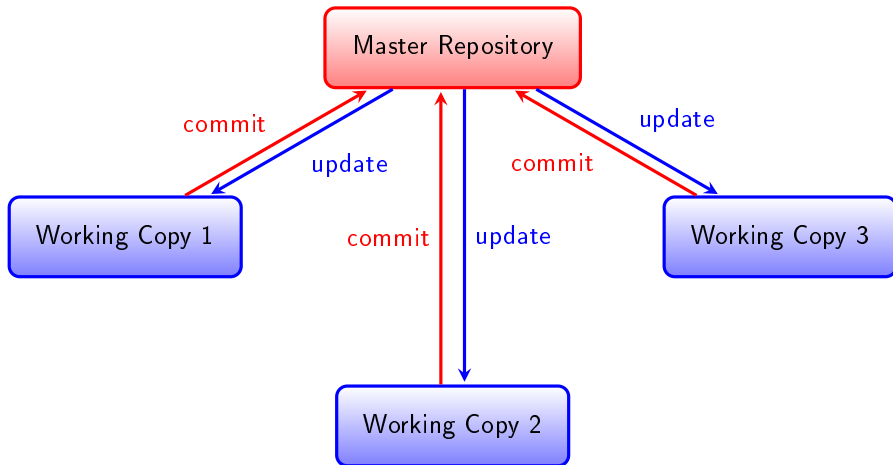
- Distributed Versus Centralised

Collaborative Software Development Workflows

Bazaar Basics

Bazaar in an SVN environment

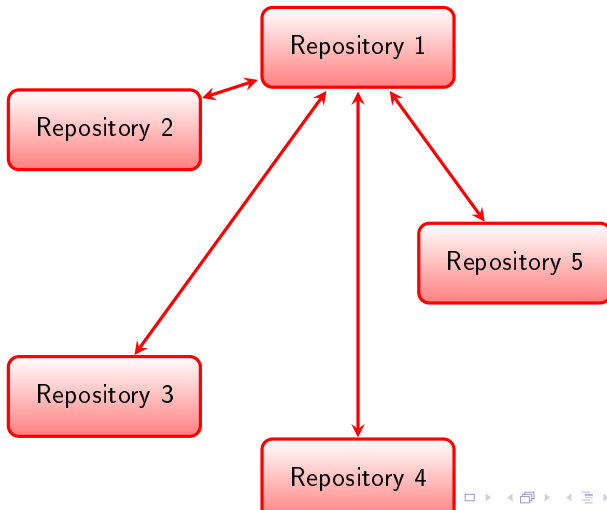
Central Repository



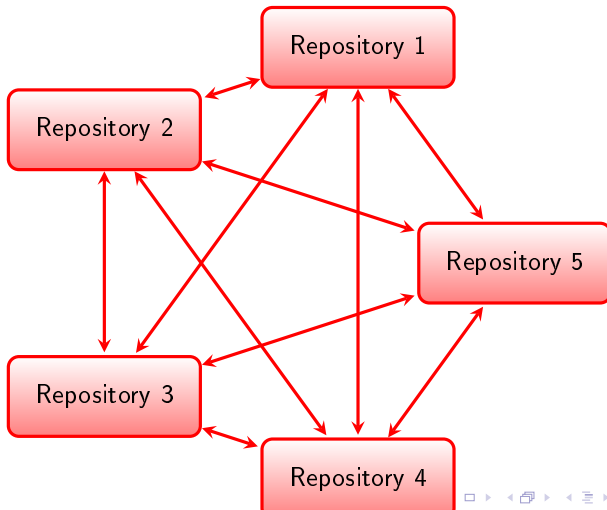
Centralised Version Control Terminology

- Repository** central store of data under revision control
- Working copy** ordinary directory tree on local system with an administrative directory
- Revision** a state of the filesystem tree in the repository
- Commit** publish files to the repository and create a new revision
- Update** apply the state of a repository revision to the working copy
- Merge** resolve conflicting differences after an update

Distributed Branches



Distributed Branches



Distributed Version Control Terminology

Revision a snapshot of the files you're working with

Working tree the directory containing your version-controlled files and sub-directories

Branch an ordered set of revisions that describe the history of a set of files

Repository a store of revisions

Standalone tree a working tree, branch, and repository all in a single location

Changeset difference (delta) between two revisions of a branch

Pull transfer changesets from a remote branch to the current branch

Push transfer changesets from the current branch to a remote branch

Merge apply changesets from a remote divergent branch to reconcile the divergent branches

Distributed Versus Centralized

Centralised

- ▶ canonical revisions
- ▶ working copies depend on repository
- ▶ commits visible to everyone
- ▶ no relationship between working copies
- ▶ all changes go through repository
- ▶ can work on sub trees of the repository

Distributed

- ▶ changes instead of versions
- ▶ standalone trees are completely independent
- ▶ commits local to the branch
- ▶ branches keep reference to their ancestors
- ▶ move changes freely between branches
- ▶ branches are atomic

Outline

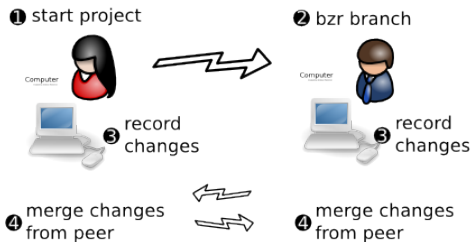
Version Control System Concepts

Collaborative Software Development Workflows

Bazaar Basics

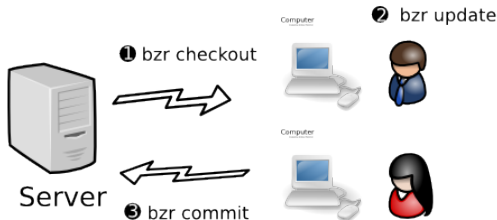
Bazaar in an SVN environment

Working with a Peer: The Partner Workflow



http://doc.bazaar.canonical.com/bzr.dev/en/_images/workflows_peer.png

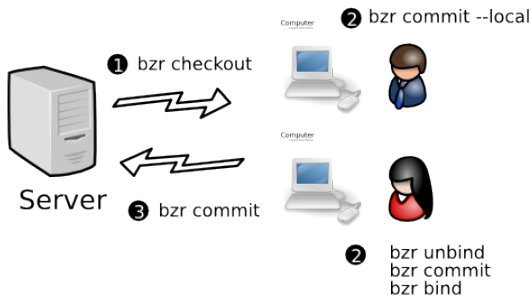
Team Collaboration: Centralised Repository (SVN style)



http://doc.bazaar.canonical.com/bzr.dev/en/_images/workflows_centralized.png

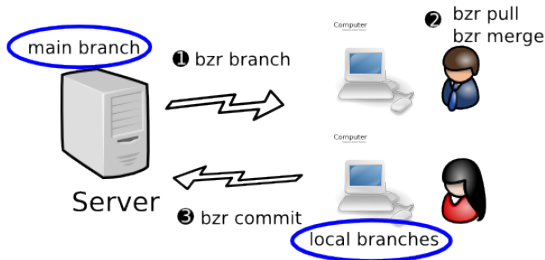


Working Offline with a Centralised Repository



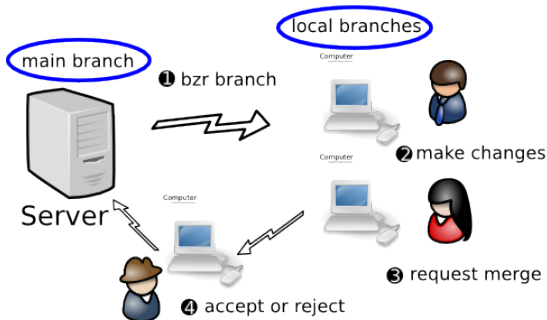
http://doc.bazaar.canonical.com/bzip.dev/en/_images/workflows_localcommit.png

Team Collaboration: Distributed Development with Shared Mainline



http://doc.bazaar.canonical.com/bzr.dev/en/_images/workflows_shared.png

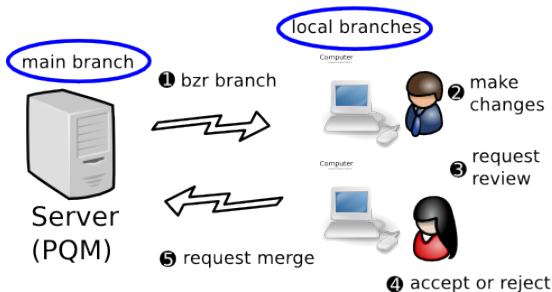
Team Collaboration: Decentralised with Human Gatekeeper



http://doc.bazaar.canonical.com/bzr.dev/en/_images/workflows_gatekeeper.png



Team Collaboration: Decentralised with Automatic Gatekeeper



http://doc.bazaar.canonical.com/bzr.dev/en/_images/workflows_pqm.png

Outline

Version Control System Concepts

Collaborative Software Development Workflows

Bazaar Basics

Bazaar in an SVN environment

Bazaar Version Control System (bzd)

<http://www.bazaar-vcs.org>

- ▶ Distributed version control
- ▶ Open source
- ▶ Implemented in Python
- ▶ Maintained by Canonical (the company behind ubuntu)
- ▶ Hosted on launchpad

Bazaar Version Control System (bzd)

<http://www.bazaar-vcs.org>

- ▶ Distributed version control
- ▶ Open source
- ▶ Implemented in Python
- ▶ Maintained by Canonical (the company behind ubuntu)
- ▶ Hosted on launchpad

Hands-on session!

Protocols / URLs supported by Bazaar

`aftp://` Access using active FTP.

`bzr://` Fast access using the Bazaar smart server.

`bzr+ssh://` Fast access using the Bazaar smart server over SSH.

`file://` Access using the standard filesystem (default)

`ftp://` Access using passive FTP.

`http://` Read-only access of branches exported on the web.

`https://` Read-only access of branches exported on the web using SSL.

`sftp://` Access using SFTP (most SSH servers provide SFTP).

`svn://` Access using the Subversion smart server.

`svn+ssh://` Access using the Subversion smart server tunneled over SSH.

Outline

Version Control System Concepts

Collaborative Software Development Workflows

Bazaar Basics

Bazaar in an SVN environment

bzr-svn: Bridge Between Central SVN Repository and Local bzr Branch

<https://launchpad.net/bzr-svn>

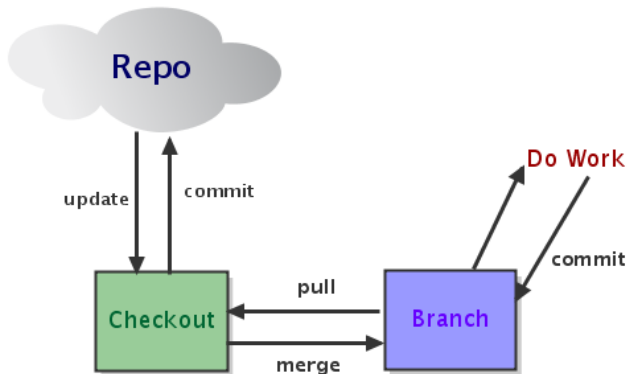
- ▶ Checkout SVN branch with bzr
- ▶ Fast offline history, logs, diffs
- ▶ Branch, commit, pull, push locally
- ▶ Commit back to SVN repository

Caution: initial checkout takes ages!

Limitations

- ▶ no support for svn:externals
- ▶ some subversion properties are not supported (svn:ignore, svn:keywords, etc.)

bzr-svn: Workflow



http://www.lornajane.net/uploads/tech/bzr_checkout_and_branch.png

Migrating SVN repositories

`bzr svn-import` converts an SVN repository to a bzr repository (+ branch)

- ▶ works incremental (i.e. only pulls new revision when applied to an existing bzr branch)
- ▶ can be set up to mirror an SVN repository (e.g. via SVN hooks)

`svn2bzr` (statically) converts an SVN dump to a bzr repository (+ branch)